

Môn thi: Cấu trúc dữ liệu và giải thuật

Mã lớp: Các lớp đại trà, chất lượng cao

Thời gian làm bài: 90 phút

(Sinh viên không được sử dụng tài liệu)

Thang điểm gợi ý :

Câu	Nội dung	Điểm
1.a	Cho biết độ phức tạp của thuật toán Insertion sort	0.25 điểm
1.b	Viết đúng hàm sắp xếp mảng 1 chiều N phần tử giảm dần theo thuật toán insertion sort	0.75 điểm
1.c	Mô tả dãy số thay đổi qua từng bước thuật toán	1 điểm
2.a	Vẽ cây nhị phân tìm kiếm đúng	1 điểm
2.b	Duyệt cây theo RNL, NRL	1 điểm
2.c	Huỷ lần lượt từng nút L, T, E, R trên cây	1 điểm
3.1.a	Xác định các khóa khi thêm vào cây sẽ làm phát sinh thao tác tách node	0.5 điểm
3.1.b	Vẽ cây B-Tree trước và sau khi thêm các khóa ở 3.1.a	1 điểm
3.2	Xóa các khóa khỏi cây: 13, 24, 19 và vẽ cây B-Tree trước và sau khi xóa mỗi khóa	0.5 điểm
4.a	Trình bày từng bước việc tìm mã quản lý 23 trong bảng băm	0.5 điểm
4.b	Trình bày từng bước việc thêm các mã quản lý 11, 20, 27 vào bảng băm theo đúng thứ tự	1.5 điểm
5.a	Định nghĩa CTDL phù hợp	0.5 điểm
5.b	Viết hàm nhập theo Input và lưu trữ thông tin của đồ thị vào cấu trúc dữ liệu ở 5.a	0.5 điểm

Tổng điểm : 10

Câu 1:

- a. Hãy cho biết độ phức tạp của thuật toán Insertion sort (chèn trực tiếp) theo định nghĩa Big-O (O lớn) (0.25 điểm)

Đáp án tham khảo:

Độ phức tạp thuật toán trong 3 trường hợp: tốt nhất, xấu nhất, TB là $O(n^2)$

- b. Viết hàm sắp xếp mảng 1 chiều gồm N phần tử giảm dần với thuật toán Insertion sort (0.75 điểm)

Đáp án tham khảo:

<pre>void InsertionSort(int a[], int n) { for(int i=1;i<=n-1;i++) { int x = a[i]; for(int j=i-1; j>=0&&a[j]<x;j--) a[j+1]=a[j]; a[j+1] = x; } }</pre>	<p>// đúng vòng lặp for/while thứ nhất :0.25đ</p> <p>// đúng vòng lặp for/while thứ hai: 0.25đ</p> <p>//đúng phép gán 0.25đ</p>
--	---

- c. Hãy cho biết dãy số sẽ thay đổi qua từng bước như thế nào khi áp dụng thuật toán ở câu 1b, biết rằng dãy số cho như sau: 3, 8, 4, 5, 9, 1, 2, 6 (1 điểm)

Đáp án tham khảo:

Bước	Mảng cần sắp xếp	Ghi chú
1	8 3 4 5 9 1 2 6	Chèn 8 đúng – 0.25 điểm
2	8 4 3 5 9 1 2 6	Chèn 4 đúng – 0.125 điểm
3	8 5 4 3 9 1 2 6	Chèn 5 đúng – 0.125 điểm
4	9 8 5 4 3 1 2 6	Chèn 9 đúng – 0.125 điểm
5	9 8 5 4 3 1 2 6	Chèn 1 đúng – 0.125 điểm
6	9 8 5 4 3 2 1 6	Chèn 2 đúng – 0.125 điểm
7	9 8 6 5 4 3 2 1	Chèn 6 đúng – 0.125 điểm

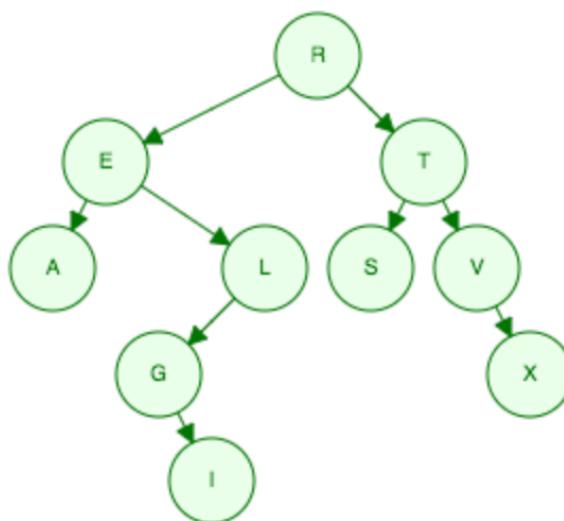
Câu 2:

Cho dãy ký tự như sau: R, E, T, A, V, X, L, G, S, I

Hãy thực hiện các yêu cầu sau:

- a. Vẽ cây nhị phân tìm kiếm bằng cách thêm lần lượt từng ký tự vào cây theo thứ tự từ trái qua phải của dãy ký tự trên, biết rằng giá trị của từng ký tự tương ứng theo thứ tự xuất hiện của ký tự trong từ điển (1 điểm)

Đáp án tham khảo:



- b. Cho biết kết quả duyệt cây theo RNL, NRL (1 điểm)

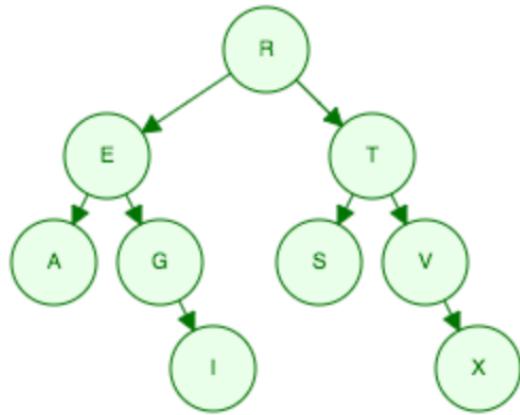
Đáp án tham khảo:

- Duyệt cây RNL – 0.5 điểm: X,V,T,S,R,L,I,G,E,A
- Duyệt cây NRL – 0.5 điểm: R,T,V,X,S,E,L,G,I,A

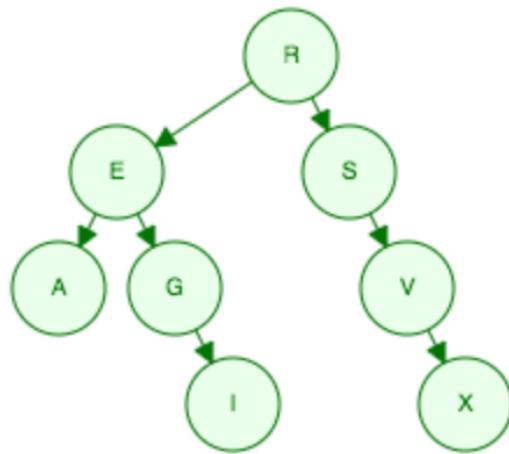
- c. Huỷ lần lượt từng nút L, T, E, R trên cây, mỗi lần huỷ 1 nút vẽ lại cây nối tiếp theo như thứ tự huỷ (1 điểm)

Đáp án tham khảo:

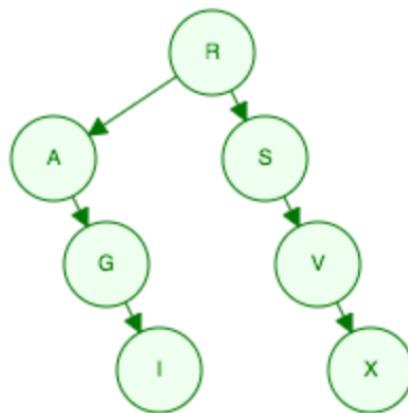
- Xoá L – 0.25 điểm



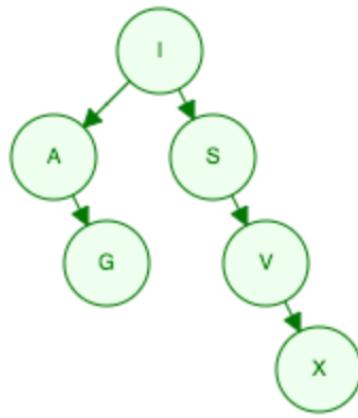
- Xoá T – 0.25 điểm



- Xoá E – 0.25 điểm



- Xoá R – 0.25 điểm



Câu 3:

Cho biết cây B-Tree bậc 3 là một cây thỏa mãn các tính chất sau:

- Tất cả node lá nằm trên cùng một mức
- Tất cả các node, trừ node gốc và node lá, có ***tối thiểu*** 2 node con.
- Tất cả các node có ***tối đa*** 3 con
- Tất cả các node, trừ node gốc, có từ 1 cho đến 2 khóa (keys)
- Một node không phải lá và có n khóa thì phải có n+1 node con.

Hãy thực hiện các yêu cầu sau:

3.1 Cho dãy số: 12, 17, 20, 23, 15, 11, 24, 13, 19, 22, 18, 21, 16. Hỏi khi lần lượt thêm các số trong dãy theo thứ tự từ trái qua phải vào một cây B-Tree bậc 3 rỗng thì:

- a. Các khóa nào khi thêm vào cây sẽ làm phát sinh thao tác tách (split) node? (0.5 điểm)

Đáp án tham khảo:

Các khóa làm phát sinh thao tác split node là: 20, 11, 24, 22, 16 (0.5 điểm)

sinh viên có thể ghi không theo thứ tự trên nhưng đảm bảo chính xác giá trị của 05 khóa trên thì vẫn được 0.5 điểm

- b. Vẽ cây B-Tree trước và sau khi thêm các khóa ở câu a (1 điểm)

Đáp án tham khảo:

Sinh viên không cần vẽ chi tiết từng bước thực hiện thao tác split, chỉ cần vẽ trạng thái cây trước khi tiến hành thêm khóa và cây sau khi thêm khóa vào hoàn tất. Tổng cộng có 10 hình cần vẽ. Ở câu này sinh viên bắt buộc phải làm theo thứ tự các khóa sẽ được thêm. Khi vẽ sai ở bước nào thì các bước sau sẽ không được điểm.

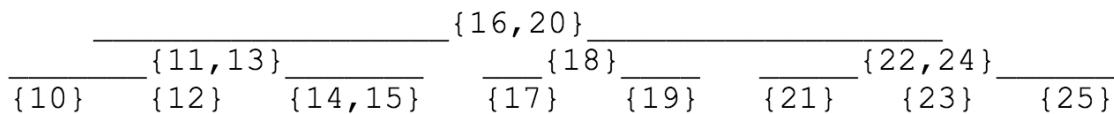
Khóa	trước	Sau
20	{12, 17}	$\overline{\{17\}}$ $\overline{\{12\}} \quad \overline{\{20\}}$
11	$\overline{\{17\}}$ $\overline{\{12, 15\}} \quad \overline{\{20, 23\}}$	$\overline{\{12, 17\}}$ $\overline{\{11\}} \quad \overline{\{15\}} \quad \overline{\{20, 23\}}$
24	$\overline{\{12, 17\}}$ $\overline{\{11\}} \quad \overline{\{15\}} \quad \overline{\{20, 23\}}$	$\overline{\{17\}}$ $\overline{\{12\}}$ $\overline{\{15\}}$ $\overline{\{20\}}$ $\overline{\{23\}}$ $\overline{\{24\}}$
22	$\overline{\{12\}}$ $\overline{\{17\}}$ $\overline{\{11\}}$ $\overline{\{13, 15\}}$ $\overline{\{23\}}$ $\overline{\{19, 20\}}$ $\overline{\{24\}}$	$\overline{\{17\}}$ $\overline{\{12\}}$ $\overline{\{13, 15\}}$ $\overline{\{19\}}$ $\overline{\{20, 23\}}$ $\overline{\{22\}}$ $\overline{\{24\}}$
16	$\overline{\{12\}}$ $\overline{\{17\}}$ $\overline{\{11\}}$ $\overline{\{13, 15\}}$ $\overline{\{20, 23\}}$ $\overline{\{18, 19\}}$ $\overline{\{21, 22\}}$ $\overline{\{24\}}$	$\overline{\{17\}}$ $\overline{\{12, 15\}}$ $\overline{\{13\}}$ $\overline{\{16\}}$ $\overline{\{18, 19\}}$ $\overline{\{20, 23\}}$ $\overline{\{21, 22\}}$ $\overline{\{24\}}$

Thang điểm:

Vẽ đúng TOÀN BỘ từ trước khi thêm khóa 20 đến sau khi thêm khóa 24: 0.5 điểm

Vẽ đúng hình cả trước và sau khi thêm khóa 22: 0.25 điểm, đúng hình cả trước và sau khi thêm khóa 16: 0.25 điểm

3.2 Cho cây B-Tree bậc 3 như hình sau:



Hãy lần lượt tiến hành xóa các khóa sau khỏi cây: 13, 24, 19 và vẽ cây B-Tree trước và sau khi xóa mỗi khóa trên (0.5 điểm)

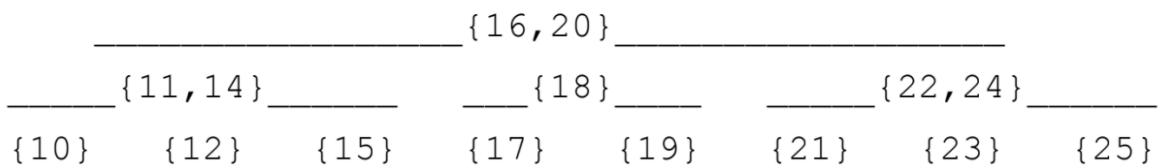
Lưu ý khi xoá:

- Khi khóa cần xóa (gọi là x) không nằm ở node lá, chọn khóa thế mạng là khóa có giá trị lớn nhất mà nhỏ hơn x.
- Thao tác nhường khoá (underflow) sẽ được thực hiện khi hai node liền kề có tổng số khóa ≥ 2 . Khi có một node không còn đáp ứng đủ số lượng khóa tối thiểu, ưu tiên thực hiện underflow thay cho catenation (hợp) vì thao tác này không làm thay đổi số khóa của node cha.
- Khi có 02 lựa chọn node liền kề để thực hiện catenation, ưu tiên chọn catenation giữa node bị thiếu khóa với node liền trước.

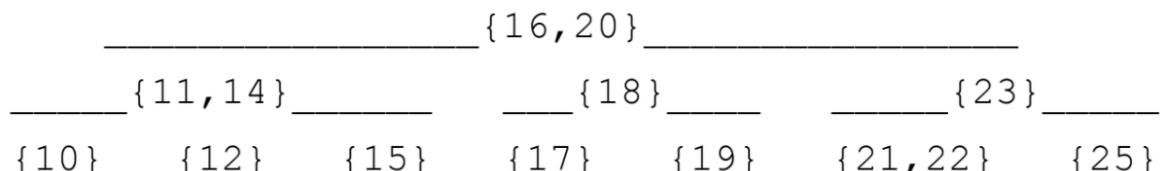
Đáp án tham khảo:

Câu này bắt buộc phải làm theo thứ tự xóa, nếu xóa sai ở khóa nào thì các bước tiếp theo chắc chắn xem như sai. Đề thi đã quy định rõ các quy ước cần tuân theo khi xóa nên chỉ có duy nhất 01 đáp án đúng có thể được chấp nhận.

- Xóa 13: swap 13 và 12 sau đó xóa. Ưu tiên thực hiện underflow giữa node {14, 15} và node vừa bị thiếu khóa thay vì catenate nó với node {10}. Xóa đúng được 0.125 điểm



- Xóa 24 - Swap 24 với 23 và thực hiện catenate, ưu tiên catenate với {21} thay vì {25}. Xóa đúng được 0.125 điểm



- Xóa 19 trực tiếp tại lá sau đó xóa và catenate node bị xóa với {17} làm {18} bị thiếu khóa và lan truyền xử lý này lên khiến node {11, 14} phải underflow cho node bị thiếu khóa. Xóa đúng được 0.25 điểm

Câu 4:

Để việc tìm kiếm thông tin mặt hàng được nhanh chóng, người ta dùng một bảng băm theo phương pháp thăm dò, làm việc trên mã quản lý của mặt hàng. Mã quản lý này là một con số nguyên. Bảng băm có:

- Hàm băm: $h(\text{key}) = (\text{key} \% M)$
- Hàm băm lại (hàm thăm dò): $\text{prob}(\text{key}, i) = (h(\text{key}) + i*i + i) \% M$

Trong đó:

- key là giá trị khóa.
- i là một số nguyên cho biết lần băm lại (thăm dò) thứ i.
- M là kích thước bảng băm.

Giả sử $M = 7$, cho trường hợp T của bảng băm đã chứa dữ liệu như bên dưới. Biết “-” là ký hiệu vị trí trống trong bảng băm.

Bảng băm T

0	-
1	-
2	16
3	-
4	-
5	12
6	13

- a. Trình bày từng bước việc tìm mã quản lý 23 trong bảng băm T . (0.5 điểm)

Đáp án tham khảo:

- Tính hàm băm với khóa 23, được vị trí 2 đang chứa giá trị 16 khác 23 \rightarrow băm lại (0.25 điểm cho việc so sánh khóa và băm lại)
- Băm lại lần 1, được vị trí 4 đang chứa giá trị “-” \rightarrow không tìm thấy khóa 23 (0.25 điểm cho việc xác định kết quả không tìm thấy khi gặp vị trí trống)

- b. Trình bày từng bước việc thêm các mã quản lý sau vào bảng băm T theo đúng thứ tự liệt kê là 11, 20, 27 (1.5 điểm).

Đáp án tham khảo:

- Thêm khóa 11:
- Tính hàm băm với khóa 11, được vị trí 4 đang chứa giá trị “-” \rightarrow Thêm 11 vào vị trí 4 (0.25 điểm cho việc tính đúng vị trí 4 và 0.25 điểm cho thao tác thêm khi gặp vị trí trống)

- Bảng băm T

0	-
1	-
2	16
3	-

4	11
5	12
6	13

- Thêm khóa 20:

- Tính hàm băm với khóa 20, được vị trí 6, dung độ → băm lại (**0.25 điểm cho việc xác định dung độ và băm lại**)
- Băm lại lần 1, được vị trí 1 đang chứa giá trị “-“, → thêm 20 vào vị trí 1 (**0.25 điểm cho việc tính đúng vị trí và thêm khi gặp vị trí trống**)
 - Bảng băm **T**

0	-
1	20
2	16
3	-
4	4
5	12
6	13

- Thêm khóa 27:

- Tính hàm băm với khóa 27, được vị trí 6, dung độ → băm lại
- Băm lại lần 1, được vị trí 1, dung độ → băm lại
- Băm lại lần 2, được vị trí 5, dung độ → băm lại
- Băm lại lần 3, được vị trí 4, dung độ → băm lại
- Băm lại lần 4, được vị trí 5, dung độ → băm lại
- Băm lại lần 5, được vị trí 1, dung độ → băm lại
- Băm lại lần 6, được vị trí 6, dung độ → băm lại (**0.25 điểm cho việc thực hiện băm lại 6 lần**)
- Đã băm lại M-1 = 6 lần → kết luận không thêm được / không tìm được vị trí thêm / không tìm được chỗ trống (**0.25 điểm cho việc không thêm được khi băm lại M-1 lần**)

Câu 5:

Trong các ứng dụng thực tế, chẳng hạn trong mạng lưới giao thông đường bộ, đường thủy hoặc đường hàng không, người ta không chỉ quan tâm đến việc tìm đường đi giữa hai địa điểm mà còn phải lựa chọn một hành trình tiết kiệm nhất (theo tiêu chuẩn không gian, thời gian hay chi phí). Vấn đề này có thể được mô hình hóa thành một bài toán trên đồ thị, trong đó mỗi địa điểm được biểu diễn bởi một đỉnh, cạnh nối hai đỉnh biểu diễn cho “đường đi trực tiếp” giữa hai địa điểm (tức không đi qua địa điểm trung gian) và trọng số của cạnh là khoảng cách giữa hai địa điểm.

Bài toán có thể phát biểu dưới dạng tổng quát như sau: Cho một đơn đồ thị có hướng và có trọng số dương $G=(V,E)$, trong đó V là tập đỉnh, E là tập cạnh (cung) và các cạnh đều có trọng số, hãy tìm một đường đi (không có đỉnh lặp lại) ngắn nhất từ đỉnh xuất phát S thuộc V đến đỉnh đích F thuộc V .

Giả sử thông tin đầu vào của bài toán (Input) được nhập vào chương trình như sau:

Input	Giải thích
7	- Dòng đầu tiên chứa một số nguyên dương e cho biết số cạnh của đồ thị
A B 1	- Với e dòng tiếp theo, mỗi dòng chứa hai chuỗi u, i và một số nguyên dương x , thể hiện thông tin có một cạnh nối từ đỉnh u sang đỉnh i trong đồ thị với độ dài (trọng số) là x
B E 3	
E D 3	

C B 4	- Dòng cuối cùng chứa hai chuỗi s và f, đây là đỉnh bắt đầu và đỉnh kết thúc của đường đi cần tìm
A D 7	
E C 2	
C D 1	Lưu ý: không biết trước số đỉnh và danh sách các đỉnh.
A E	

Hãy thực hiện các yêu cầu sau:

- Xây dựng các cấu trúc dữ liệu phù hợp nhất có thể để biểu diễn đồ thị trên máy tính theo input đã cho. (0.5 điểm)
Cấu trúc được xem là tốt nếu đạt được các tiêu chuẩn sau: Tiết kiệm tài nguyên; Hỗ trợ một số thao tác cơ bản như “Kiểm tra hai đỉnh có kề nhau không”, “Tim danh sách các đỉnh kề với một đỉnh cho trước” với ràng buộc là không phải duyệt qua danh sách tất cả các cạnh của đồ thị.
- Viết hàm nhập theo Input ở đầu bài và lưu trữ thông tin của đồ thị vào cấu trúc dữ liệu đã đề xuất ở câu a. (0.5 điểm)

*** KHÔNG YÊU CẦU tìm cách giải cho bài toán này. Sinh viên ĐƯỢC PHÉP sử dụng Standard Template Library-STL với những cấu trúc dữ liệu (vector, stack, queue, list, map, set, pair, ...) cũng như giải thuật được xây dựng sẵn.

Đáp án tham khảo:

I. Một số lưu ý:

1. Đây là câu hỏi có tính thử thách cao để phân loại sinh viên giỏi.

Nếu như “Bài toán tìm đường đi” chưa được giới thiệu trên lớp/hoặc không có trong slide bài giảng chung thì cũng không vấn đề gì, vì đề bài có mô tả bài toán rõ ràng, không yêu cầu SV tìm cách giải, đề thi kiểm tra về “cách biểu diễn đồ thị trên máy tính, ưu và nhược điểm của mỗi phương pháp”.

2. Việc chọn cấu trúc dữ liệu nào để biểu diễn đồ thị phụ thuộc vào từng tình huống cụ thể và có tác động rất lớn đến hiệu quả của thuật toán.

Để tránh trường hợp SV “sáng tạo” những cách biểu diễn không phù hợp như lưu toàn bộ input thành chuỗi, hoặc tạo nhiều biến/mảng không cần thiết, đề bài có ghi rõ tiêu chuẩn đánh giá. Nếu không có tiêu chuẩn rõ ràng thì GV rất khó chấm điểm.

3. Khi trình bày CTDL cần làm rõ: khai báo kiểu dữ liệu mô tả cho các đối tượng thông tin trong bài toán; Cho biết ý nghĩa, vai trò của từng biến, mảng, cấu trúc, thuộc tính của cấu trúc, lớp và thuộc tính của lớp.... được sử dụng → Nếu SV không giải thích gì, người chấm đọc không hiểu và dễ gây nhầm lẫn thì có thể TRỪ ĐIỂM

II. CTDL phù hợp cho tình huống đã đặt ra

Input không cho biết trước số đỉnh và danh sách các đỉnh và trên thực tế đồ thị cho bài toán này thường sẽ thưa (sparse). Cách tổ chức đồ thị hợp lý nhất là danh sách kề. Danh sách kề có rất nhiều cách cài đặt. Nếu SV dùng danh sách kề, trong đó mô tả CTDL rõ ràng, viết hàm nhập chấp nhận được thì trọn 1 điểm

III. Các cách biểu diễn khác

Một số cách biểu diễn khác dù không đáp ứng đầy đủ các tiêu chuẩn trong đề bài nhưng vẫn có thể chấp nhận và cho điểm, **nhưng chỉ được tối đa 0.5/1 điểm**

1. Dùng ma trận kề, ma trận trọng số

Cách biểu diễn này cần biết trước số đỉnh (để khai báo kích thước ma trận) và danh sách các đỉnh (để có thể ánh xạ từ mỗi đỉnh sang index dòng/cột tương ứng trong ma

trận). Nếu khai báo kích thước ma trận lớn gây lãng phí bộ nhớ (đặc biệt khi đồ thị thưa)

2. Dùng danh sách cạnh

Nhược điểm là để xác định những đỉnh nào của đồ thị là kề với một đỉnh cho trước chúng ta phải duyệt qua danh sách tất cả các cạnh của đồ thị

Bảng thang điểm gợi ý chi tiết cho 2 câu 5.a và 5.b

Cách thực hiện sử dụng :	Mô tả	Thang điểm Thiết kế CTDL (5.a)	Thang điểm Hàm nhập theo input (5.b)
Danh sách kề	<p>Ứng với mỗi đỉnh i, ta cần lưu trữ một tập hợp (danh sách) gồm các đỉnh kề với đỉnh i</p> <p>Có nhiều cách cài đặt</p> <p>Ví dụ: Cài đặt Danh sách kề dùng map trong STL</p> <pre>map<string, set<pair<string, int>> adj_list</pre> <p>Có thể dùng vector, list, tree thay cho set cũng được.</p> <pre>A → {(B, 1), (D, 7)} B → {(E, 3)}</pre> <p>Hoặc</p> <pre>map<string, map<string, int>> adj_list</pre> <pre>A--> [B-->10] [O-->45] [C-->12] B--> [C-->4] [L-->8]</pre> <p>Ví dụ: Cài đặt Danh sách kề dùng các danh sách liên kết</p> <pre>vector<list<node>> adj_list</pre> <p>và 1 cấu trúc để tra cứu từ chuỗi sang index. Struct node chứa thông tin tên đỉnh và trọng số.</p>	0.5 điểm	0.5 điểm
Ma trận trọng số	<ul style="list-style-type: none"> Giá trị của mỗi ô trong ma trận nếu khác 0 thì cho biết trọng số của cạnh nối giữa 2 đỉnh tương ứng Do đỉnh có tên là chuỗi nên cần có cách ánh xạ từ chuỗi sang số nguyên cho biết index dòng/cột tương ứng trong ma trận, nếu SV không xử lý được thì không có điểm <p>Ví dụ: cách ánh xạ từ chuỗi sang index</p> <ol style="list-style-type: none"> map<string, int> name_to_index; 		

	<p>hoặc</p> <p>2. dùng mảng lưu các đỉnh (khi muốn biết đỉnh tương ứng với index nào thì phải search)</p>		
	<p>Nếu SV có cách xử lý đếm được số đỉnh của đồ thị và khai báo kích thước ma trận phù hợp thì có thể đạt tối đa 1 điểm</p> <p>Ví dụ:</p> <p>Cách 1</p> <pre>vector<string> nodes; vector<pair<string, string>> edges;</pre> <p>Khi nhập cạnh thì kiểm tra 2 đỉnh có trong nodes chưa, nếu chưa thì thêm đỉnh mới vào và thêm cạnh vào edges</p> <pre>int nums=nodes.size() vector<vector<int>> matrix (num, vector<int>(num,0))</pre> <p>Cách 2: cấp phát động và thêm dòng, thêm cột khi có nhu cầu</p>	0.5 điểm	0.5 điểm
	<p>Nếu chỉ define MAX 2000 thì kích thước không phù hợp, chỉ đạt tối đa 0.5</p> <pre>vector <vector<int> > matrix (Max, vector<int> (Max, 0)); hoặc int **matrix; hoặc int matrix[MAX][MAX]</pre>	0.25 điểm	0.25 điểm
Danh sách cạnh	<p>Lưu các bộ 3 (x,y,w) tương ứng điểm đầu, điểm cuối và trọng số của các cạnh.</p> <p>Ví dụ:</p> <p>1. map<pair<string, string>, int> (A,B)-->1 (A,C)-->2 (A,D)-->4</p> <p>Hoặc</p> <p>2. Dùng 1 ma trận có m dòng 2 cột lưu m cạnh và thêm 1 mảng 1 chiều khác lưu m trọng số</p> <p>Hoặc</p> <p>3. Dùng mảng/danh sách... kiểu node, mỗi node có 3 thuộc tính (x,y,w) ...</p>	0.25 điểm	0.25 điểm

Hết